Pontificia Universidad
JAVERIANA
Cali

Detección de Noticias Falsas: comparación entre modelos de aprendizaje profundo basados en redes neuronales y modelos de lenguaje de gran escala

Programa de Maestría en Ingeniería

Presentado por:

CLAUDIA PATRICIA OVIEDO SANTACRUZ

Dirigido por:

Julián Gil González Ph.D.

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Junio de 2025

Detección de Noticias Falsas: comparación entre modelos de aprendizaje profundo basados en redes neuronales y modelos de lenguaje de gran escala

C. P. Oviedo
Coviedo11@javerianacali.edu.co, Facultad de Ingeniería, Pontificia Universidad Javeriana, Calle 18 No. 118-250. Cali, Colombia

Abstract

En la era digital actual, la propagación de noticias falsas en plataformas digitales representa un desafío creciente para la sociedad, debido a su potencial para desinformar y generar consecuencias sociales, políticas y económicas. Este trabajo compara el desempeño de dos enfoques de aprendizaje profundo en la detección automática de noticias falsas: un modelo basado en redes neuronales Long Short-Term Memory y un modelo de lenguaje de gran escala preentrenado como BERT. Se utilizaron conjuntos de datos abiertos y técnicas de preprocesamiento de texto, además de estrategias de sintonización de hiperparámetros a, para optimizar el rendimiento de cada modelo. Los resultados muestran que el modelo Long Short-Term Memory, tras la optimización, alcanzó una precisión del 92%, superando al modelo de lenguaje de gran escala, que logró un 89%. Estos hallazgos evidencian que, en tareas específicas y bajo condiciones controladas, modelos más livianos y tradicionales pueden superar a modelos más complejos, reafirmando la importancia de una selección cuidadosa del modelo en función del problema a tratar y los recursos disponibles.

1. Introducción

El uso de internet ha crecido exponencialmente en los últimos años, convirtiéndose en una herramienta esencial para una amplia variedad de actividades y tareas. Un informe anual sobre redes sociales y tendencias digitales [1] reveló que, para enero de 2025, Colombia contaba con 41.1 millones de usuarios del internet. En promedio, los colombianos pasan 8 horas y 44 minutos en línea diariamente, de los cuales 3 horas y 57 minutos se destinan a redes sociales como WhatsApp, Instagram y Facebook.

Esto posiciona al país como el sexto a nivel mundial donde las personas invierten más tiempo en estas plataformas, lo que a su vez amplifica el riesgo de exposición a contenidos falsos.

El auge de las redes sociales ha facilitado la comunicación, la conexión entre personas y la difusión masiva de información. Sin embargo, gran parte de la información que circula en medios digitales se considera dudosa o intencionalmente falsa, lo que se conoce como "fake news". La propagación de este tipo de contenido puede generar desinformación, afectar la opinión pública y tener consecuencias significativas a nivel social, político y económico [2].

El reconocimiento de contenido falso plantea un desafío significativo debido a que la generación y distribución del contenido es rápida y masiva, lo que dificulta su análisis en grandes volúmenes de información [3]. Además, la diversidad de temas abordados añade complejidad a la tarea de verificación.

Los enfoques tradicionales para la detección de noticias falsas se han basado en el análisis de contenido y en la verificación manual de datos a través de plataformas como FactCheck.org y PolitiFact.com, con resultados ineficientes [4].

Por otro lado, con el auge del aprendizaje automático y la inteligencia artificial, la detección automática de noticias falsas se ha convertido en un campo de investigación clave para desarrollar sistemas capaces de identificar con alta precisión y en tiempo real contenido engañoso [5].

Estudios recientes han explorado diversas técnicas de extracción de características basados en frecuencia (TF-IDF), embeddings de palabras (Word2Vec, GloVe, FastText) [2,4], modelos de aprendizaje automático como los Support Vector Machines (SVM), Naive Bayes, Árboles de Decisión y Random Forest [4,6]. Además de otros modelos basados en la arquitectura Transformer como BERT, RoBERTa y ELECTRA [2,7], los cuales han obtenido buenos resultados, pero sugieren el desarrollo de técnicas de detección más eficientes y menos costosas computacionalmente que mejoren la precisión en la detección de noticias falsas.

En este trabajo se busca abordar este problema evaluando el rendimiento de dos enfoques de aprendizaje profundo para la detección de noticias falsas usando el título de la noticia. Dado que modelos menos costosos computacionalmente pueden ser competitivos bajo ciertas condiciones, se compara el rendimiento de un modelo LSTM (Long Short-Term Memory) con una arquitectura BERT (Bidirectional Encoder Representations from Transformers). Utilizando conjuntos de datos abiertos, se analizará el desempeño de estos modelos en términos de métricas como precisión, recall y F1-score. Finalmente se

optimizarán los hiperparámetros para mejorar la precisión de los modelos.

2. Métodos

La presente investigación pretende evaluar dos modelos de aprendizaje: el modelo LSTM y el modelo BERT, para luego comparar su desempeño dado un conjunto de datos. Para mejorar los resultados, se hará uso también de la sintonización de hiperparámetros. A continuación, se describe cada uno.

2.1 Modelo LSTM

Long-Short-Term Memory (LSTM) es un tipo de red neuronal recurrente ampliamente usada en aprendizaje profundo. Su aplicación es útil en problemas con datos secuenciales como el reconocimiento del habla o el procesamiento de lenguaje natural. Funciona al procesar cada entrada basada en las entradas previas, es decir, posee una memoria de las entradas pasadas. Los modelos LSTM poseen una celda de memoria que puede conservar información durante largos periodos. Esta celda está controlada por tres puertas: la puerta de entrada, la puerta de olvido y la puerta de salida. Estas puertas determinan qué información se conserva o se descarta del modelo [8].

2.2 Modelo BERT

El modelo BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje diseñado para pre entrenar representaciones bidireccionales a partir de texto sin etiquetar de izquierda a derecha y de derecha a izquierda en todas sus capas. Dado que los transformadores pueden procesar datos bidireccionales permiten que el modelo sea pre entrenado con cantidades masivas de datos. De esta manera, el modelo BERT puede ajustarse para crear modelos en una amplia gama de tareas, como respuesta a preguntas y clasificación de texto, sin modificaciones substanciales en su arquitectura [9].

2.3 Sintonización de hiperparámetros

El desempeño de cada modelo de aprendizaje depende de sus hiperparámetros. Ellos controlan los algoritmos de aprendizaje o la estructura estadística del modelo. Sin embargo, en la práctica, no hay una regla particular para escoger estos hiperparámetros, a menudo son escogidos a ensayo y error o se dejan los valores por defecto, llevando a un desempeño poco óptimo [8]

La sintonización de hiperparámetros provee una solución sistemática a este problema al buscar el conjunto de hiperparámetros que más minimice el error de validación. Las técnicas TPE , la búsqueda por cuadrícula y la búsqueda

aleatoria se encuentran entre las técnicas más usadas para este propósito [10].

La búsqueda aleatoria es una técnica de sintonización de hiperparámetros que selecciona combinaciones de hiperparámetros al azar desde un espacio definido, en lugar de explorar sistemáticamente todas las combinaciones posibles, lo que permite una mejor cobertura de espacio de búsqueda cuando hay muchos hiperparámetros o los recursos son limitados [11].

TPE (Tree-structured Parzen Estimator) es una técnica de optimización Bayesiana que modela la distribución de probabilidad de las evaluaciones de desempeño para guiar la siguiente selección de prueba. En lugar de modelar directamente la función objetivo, TPE modela dos distribuciones: una para los hiperparámetros que han producido buenos resultados y otra para los que han producido malos resultados. A partir de estas, selecciona nuevas configuraciones que maximizan la evaluación del desempeño [12].

En la Sección 3.4 se puede ver en más detalle las técnicas implementadas.

3. Metodología

Para crear un modelo de inteligencia artificial se siguen algunos pasos definidos, tal como se puede ver en la Figura 1. A continuación se describe la metodología usada:

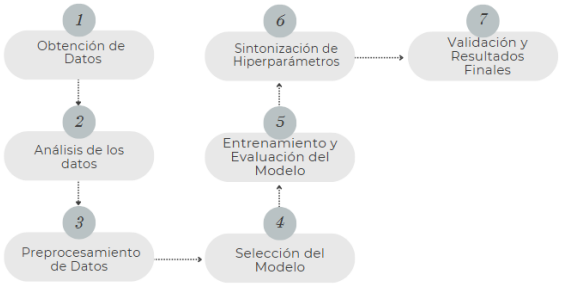


Figura 1. Metodología empelada para la detección de noticias falsas a partir de modelos de aprendizaje profundo

3.1 Obtención de datos: conociendo el corpus

Para el desarrollo de la investigación se usó el conjunto de datos abiertos “Fake News” disponible en el repositorio Kaggle [13]. El conjunto de datos contiene información de noticias clasificadas como confiables o poco confiables, además de atributos como identificador, título, autor y texto de la noticia. Cada noticia tiene una etiqueta binaria que determina la confiabilidad de la noticia. Así, una etiqueta igual a 1 indica que una noticia es poco confiable y una etiqueta igual a 0 indica que la noticia es confiable. La descripción del corpus puede verse en la Tabla 1.

Tabla 1 Descripción de la base de datos utilizada en el presente trabajo.

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

El conjunto de datos de entrenamiento se encuentra compuesto en total por 20.800 registros de noticias, con 10.413 noticias poco confiables y 10.387 confiables, mientras que el conjunto de datos de prueba consta de 5.200 registros, de los cuales 2.339 corresponden a noticias poco confiables y 2.861 a noticias confiables. Para el entrenamiento de los modelos, se empleó el título y la etiqueta de las noticias.

3.2 Analizando los datos

Con el conjunto de datos completo, se analiza la distribución de los autores para cada tipo de noticias, autores que elaboran noticias poco confiables y autores que elaboran noticias confiables, con el objetivo de encontrar diferencias significativas y se observó que:

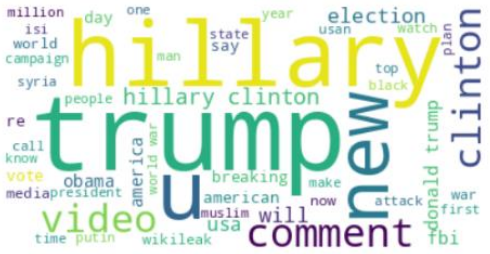
- El número total de autores diferentes es de 3.838
- El número de autores que solamente escriben noticias poco confiables es de 1.613
- El número de autores que solamente escriben noticias confiables es de 2.220
- El número de autores que escriben ambas noticias confiables o poco confiables es de 5

Solo una pequeña parte de los autores contribuyen a la elaboración de ambos tipos de noticias, lo que sugiere una polarización en los estilos de escritura que podría influir significativamente en el entrenamiento del modelo.

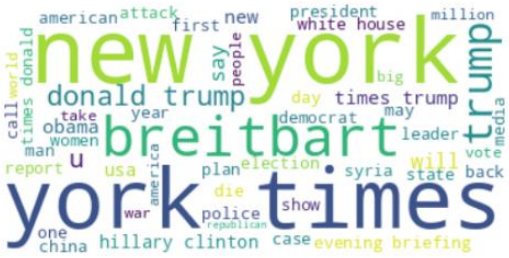
Se realizó también, la distribución de palabras comunes en títulos de ambos tipos de noticias, evidenciando las tendencias en los temas tratados en cada una, tal como se muestra en la Figura 2.

3.3 Preprocesamiento de texto

El preprocesamiento es un paso importante en los modelos de clasificación para limpiar y preparar los datos para su análisis. Algunas técnicas para el preprocesamiento incluyen limpieza de texto, manejo de contracciones y transformaciones léxicas [14]. En la Sección 4.1, se describen las técnicas usadas y los resultados obtenidos para el preprocesamiento del conjunto de datos seleccionado.



(a)



(b)

Figura 2: Comparación entre palabras más comunes en títulos de noticias poco confiables(a) y noticias confiables (b)

3.4 Selección del modelo y Sintonización de hiperparámetros

Como modelos base se seleccionaron las siguientes arquitecturas:

- Un modelo secuencial LSTM con 10 neuronas, una función de activación sigmoide y una capa de salida con clasificación binaria 1 ó 0, para establecer la predicción de que una noticia resultante pueda ser poco confiable o confiable.
- Un modelo BERT preentrenado, el cual fue ajustado al dominio de las noticias usando fine-tuning sobre el conjunto de datos.

En el primer paso, se llevó a cabo un entrenamiento con los modelos base con el fin de obtener una referencia en cuanto al rendimiento de los modelos.

Una vez obtenido los resultados preliminares del modelo LSTM, se buscó encontrar los valores de los parámetros con los cuales el modelo tiene mejor desempeño. Para ello se usó la técnica Random Search para realizar una búsqueda aleatoria de los valores óptimos. Se utilizó sobre los siguientes parámetros:

- Número de unidades LSTM: [10,50,100,200,256]
- Función de activación: [ReLU, sigmoide]
- Y tasa de aprendizaje: entre 0.01 y 0.001

Para el modelo BERT, se usó la herramienta Optuna para el afinamiento de los valores de los parámetros del modelo. Se creó un estudio para realizar la búsqueda usando un enfoque de optimización Bayesiana Eficiente, usando la técnica TPE en tres principales parámetros:

- Tasa de aprendizaje: entre 1e-5 y 5e-4.
- Tamaño de lote: [16, 32, 64]
- Número de épocas: entre 1 y 15.

La sintonización buscó maximizar la precisión en el conjunto de validación, utilizando una partición del 20 % de los

datos de entrenamiento como subconjunto de validación.

Finalmente, se seleccionó el modelo del estudio con mejores resultados de acuerdo con los parámetros óptimos encontrados.

3.5 Evaluación de los modelos

Para evaluar el rendimiento de los modelos, se usaron métricas como la exactitud (accuracy), la precisión, la puntuación F1 (F1- Score) y la sensibilidad (recall), siendo éstas algunas de las métricas más usadas en el dominio del problema [15]. A continuación, se describe brevemente cada una de las métricas usadas si tenemos en cuenta que [16] [17]:

TP (True Positives): Verdaderos positivos
TN (True Negatives): Verdaderos negativos
FP (False Positives): Falsos positivos
FN (False Negatives): Falsos negativos

Accuracy mide la proporción de predicciones correctas sobre el total de predicciones realizadas, como se ve en la ecuación (1), resulta útil cuando las clases están balanceadas.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision mide la proporción de verdaderos positivos entre todos los casos que el modelo clasificó como positivos, como se muestra en la ecuación (2). Alta precisión, indica que hay pocos falsos positivos, importante cuando el costo de un falso positivo es alto.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall mide la capacidad del modelo de detectar todos los verdaderos positivos, como se ve en la ecuación (3). Un recall alto indica que hay pocos falsos negativos, útil en casos donde los falsos negativos tienen consecuencias severas.

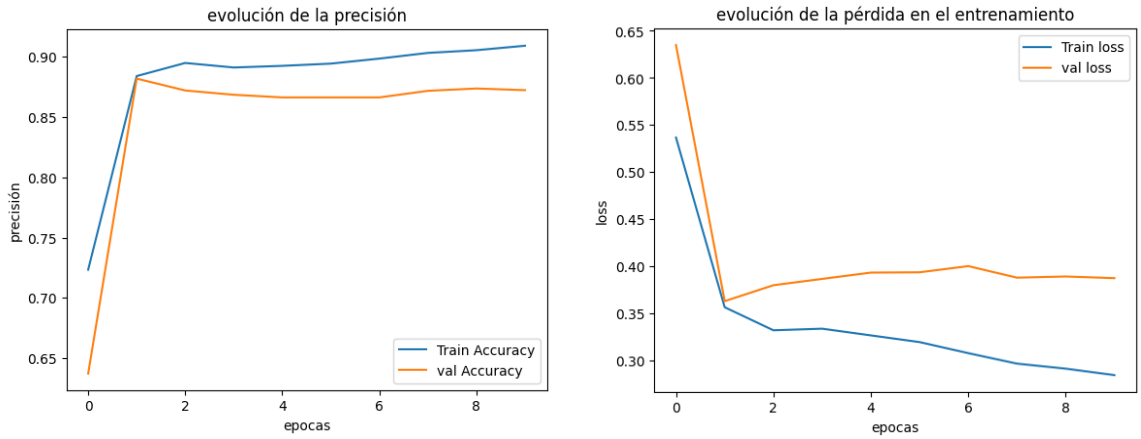
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1 Score es la media armónica entre precision y recall, proporciona un balance entre ambas métricas, como se ve en la ecuación (4). Un F1 Score alto significa que clasifica correctamente una alta proporción de positivos verdaderos (recall alto) y lo hace sin generar demasiados falsos positivos (precisión alta)

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Además de estas métricas, se graficó también la curva ROC, que muestra la relación entre dos tasas: la proporción de verdaderos positivos correctamente identificados y la proporción de falsos positivos, calculando a su vez, el área bajo la curva AUC, que proporcionará una medida del rendimiento del modelo.

La implementación de los modelos se realizó en el ambiente de ejecución Google Colab usando Python 3 en un entorno Jupyter Notebook, con una CPU



(a) *(b)*

Figura 3. Resultados del valor de precisión (a) y el valor de pérdida (b) del modelo LSTM durante el entrenamiento

Intel Xeon 2.20GHz y 22 GB de RAM. Esta implementación, asi como la evaluación y despliegue de resultados, se pueden encontrar en el repositorio de Github Deteccion-Automatica-de-Noticias-Falsas- [18]

4. Resultados

4.1 Preprocesamiento de texto

Para el preprocesamiento del conjunto de datos, se eliminó caracteres especiales, puntuaciones, números y espacios innecesarios.

Luego, el texto resultante se convirtió a minúscula y se eliminaron las palabras de parada (stopwords), que son aquellas palabras muy comunes que no aportan significado a la frase. La técnica de stemming se usó para dejar solamente la raíz acortando la palabra.

Finalmente, se usó Tokenización para dividir el texto en tokens, las unidades individuales del texto y Padding para normalizar las secuencias y hacerlas de la misma longitud. Se obtuvo entonces una representación numérica del texto que nuestro modelo puede entender.

4.2 *Entrenamiento del modelo LSTM*

Se realizó el entrenamiento del modelo LSTM en 10 épocas y se obtuvo un valor de accuracy del 86% en los datos de prueba. Del mismo modo, se obtuvo un valor de pérdida del 0.28 en los datos de entrenamiento y de 0.38 en los datos de prueba.

Se calcularon también el área bajo la curva AUC, obteniendo un valor de 0.81 y F1 Score con un valor de 0.83.

En la Figura 3 se puede observar la evolución de estos resultados del entrenamiento para las 10 épocas.

Una vez entrenado el modelo, se realizó la búsqueda de los valores de los parámetros con los cuales el modelo tuviera mejor desempeño. Se obtuvo que, con una configuración de 200 neuronas, una función de activación Sigmoide y una tasa de aprendizaje de 0.009, el modelo presenta un valor de accuracy del 92% en los datos de prueba.

Se graficó de igual forma la curva Roc para este modelo sintonizado, y se obtuvo un área bajo la curva AUC de 0.98, como se puede ver en la Figura 4.

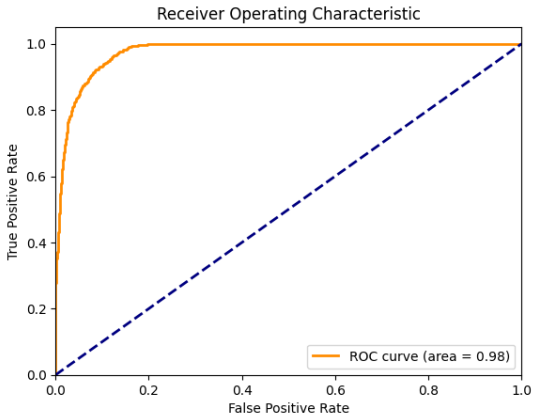


Figura 4. Curva ROC del modelo LSTM Sintonizado

4.3 Entrenamiento del modelo BERT

Se inicializó un modelo pre entrenado BERT y se realizó su entrenamiento ajustándolo al conjunto de datos del dominio del problema. Al evaluar el modelo resultante, se obtuvo un valor de accuracy del 87% en los datos de prueba.

Se graficó la curva ROC y se obtuvo también el área bajo la curva AUC, obteniendo un valor de 0.86.

Luego, se realizó un estudio para evaluar los valores de los parámetros del modelo con el objetivo de mejorar su desempeñó y se encontró que con una tasa de aprendizaje de 2e-4, con 11 épocas de entrenamiento y un tamaño de lote de 64, el modelo obtuvo un desempeño del 89% con los datos de prueba, mejorando los resultados previos obtenidos.

Se graficó la curva ROC también para este modelo BERT sintonizado y se obtuvo el área bajo la curva AUC, obteniendo un valor de 0.89, como se puede ver en la figura 5.

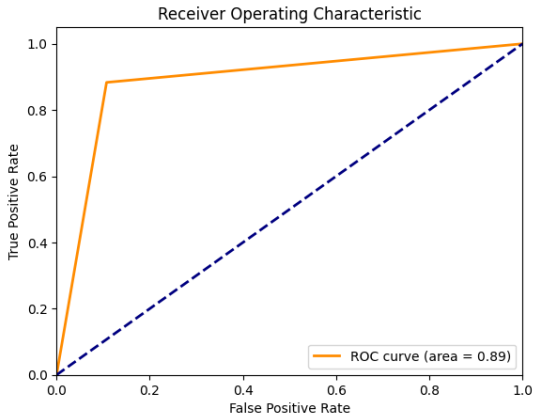


Figura 5. Curva Roc del modelo BERT sintonizado

La comparación de métricas antes y después de la Sintonización de Hiperparámetros de ambos modelos puede observarse en la Tabla 2.

Tabla 2. Comparación de Métricas de Modelos Antes y Después de la Sintonización de Hiperparámetros

Métrica	LSTM (Antes)	Bert (Antes)	LSTM (Después)	Bert (Después)
AUC-ROC	0.81	0.86	0.98	0.89
Accuracy	0.86	0.87	0.92	0.89
Precision	0.90	0.84	0.90	0.86
Recall	0.77	0.84	0.91	0.88
F1 Score	0.83	0.84	0.90	0.87

Discusión

Los resultados experimentales indican que el modelo LSTM obtuvo una precisión del **92%**, superando al BERT, que alcanzó una precisión del **89%**.

Aunque el modelo BERT tuvo un ajuste específico usando el conjunto de datos del dominio —lo cual en general, suele traducirse en mejoras sustanciales en el rendimiento del modelo en tareas especializadas—, el modelo LSTM logró mejor desempeño. Una posible explicación de este mejor desempeño es la capacidad de los modelos LSTM para modelar de forma eficiente las relaciones y dependencias secuenciales en los datos.

Resultados similares han sido reportados en estudios previos. Por ejemplo, en [19], una arquitectura basada en LSTM alcanzó una precisión del 99 %, en comparación con un 82 % obtenido por un modelo BERT en la tarea de detección de noticias falsas. De manera similar, en [20] se presenta una comparación entre una arquitectura híbrida que incluye un modelo LSTM, la cual alcanzó una precisión del 94 %, frente al 86 % obtenido por un modelo BERT en el mismo dominio.

A pesar del creciente interés en los modelos de gran escala para tareas relacionadas con el procesamiento del lenguaje natural, estos modelos suelen requerir una considerable cantidad de datos y recursos computacionales para lograr resultados competitivos en tareas especializadas.

Al comparar la cantidad de parámetros ajustables, se observa que el modelo BERT posee más de mil parámetros a ajustar en entrenamiento, lo que implica que la adaptación al dominio de interés de los modelos BERT puede resultar costosa y compleja.

Además, cabe destacar que los modelos LSTM presentan una arquitectura más compacta y tiempos de entrenamiento significativamente menores, lo cual refuerza su viabilidad como una solución

eficiente y efectiva, especialmente cuando se dispone de una cantidad limitada de datos

En conjunto, estos resultados sugieren que, a pesar del auge de los modelos de lenguaje de gran escala como los BERT, modelos más tradicionales como el LSTM siguen siendo altamente competitivos, especialmente cuando se optimizan cuidadosamente para tareas específicas.

5. Conclusiones

En este trabajo se llevó a cabo la comparación entre dos enfoques de aprendizaje profundo para la detección de noticias falsas: un modelo LSTM y un modelo BERT.

Los resultados obtenidos evidencian que, incluso tras haber sido ajustado mediante la sintonización de hiperparámetros, el modelo BERT no logró superar al modelo LSTM, que alcanzó una precisión superior del **92%** frente al **89%** obtenido por BERT. Esto sugiere que los modelos LSTM continúan siendo una solución robusta y efectiva para tareas que involucran datos secuenciales, sobre todo en contextos donde los recursos computacionales son limitados

En particular, el hecho de que un modelo LSTM más liviano haya superado a un modelo BERT ajustado refuerza la importancia de seleccionar arquitecturas adecuadas al problema en lugar de asumir que los modelos más complejos ofrecen automáticamente mejores resultados.

Como trabajo futuro se propone explorar otras estrategias para mejorar el desempeño del modelo BERT sin tener que ajustar todos los parámetros internos, como ajustar el texto de entrada (prompt engineering) y el uso de técnicas de transferencia de conocimiento.

6. Agradecimientos y reconocimientos

Deseo expresar mi agradecimiento a los profesores Julian Gil y Jorge Francisco Estela por su valiosa orientación y acompañamiento durante el desarrollo de este trabajo. De igual manera, agradezco al área de posgrados de la Facultad de Ingeniería de la Pontificia Universidad Javeriana, por haber brindado el apoyo y los recursos necesarios para su realización.

Referencias

[1] We Are Social & Meltwater, Digital 2025: Colombia, (2025),

[2] F. Farhangian, R.M.O. Cruz, G.D.C. Cavalcanti. Fake News Detection: Taxonomy and Comparative Study. Information Fusion, 103(2024) 102 - 140.

[3] X. Zhang, A.A. Ghorbani. An Overview of Online Fake News: Characterization,

Detection, and Discussion. Information Processing and Management,57 (2020) 1-26

[4] N. Capuano, G. Fenza, V. Loia, F.D. Nota. Content-Based Fake News Detection With Machine and Deep Learning: A Systematic Review. Neurocomputing, 530 (2023) 91-103.

[5] S.K. Hamed, M.J. Ab Aziz, M.R. Yaakub. A Review of Fake News Detection Approaches: A Critical Analysis of Relevant Studies and Highlighting Key Challenges Associated with the Dataset, Feature Representation, and Data Fusion. Heliyon, 9 (2023). 102140

[6] S.R. Sahoo, B.B. Gupta. Multiple Features-Based Approach for Automatic Fake News Detection on Social Networks Using Deep Learning. Applied Soft Computing Journal, 100 (2021). 106983

[7] J. Alghamdi, Y. Lin, S. Luo. Unveiling the Hidden Patterns: A Novel Semantic Deep Learning Approach to Fake News Detection on Social Media. Engineering Applications of Artificial Intelligence, 137 (2024).

[8] A. Zhang, Z. Lipton, M. Li, A. Smola, Dive into Deep learning. Cambridge University Press, 2024

[9] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, 1, 2019.

[10] H. Dabool, H. Alashwal, H. Alnuaimi, A. Alhouqani, S. Alkaabi and A. Al Ahbabi, Comparative Analysis of Hyperparameter Tuning Methods in Classification Models For Ensemble Learning, *202*4 7th International Conference on Algorithms, Computing and Artificial Intelligence *(ACAI)*, Guangzhou, China, 2024, 1-5.

[11] J. Bergstra, Y. Bengio, Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, 13 (2012)

[12] J. Bergstra, R. Bernedet, Y. Bengio, B. Kegl, Algorithms for Hyper-Parameter Optimization, Advances in Neural Information Processing Systems, 24 (2011)

[13] Conjunto de datos abiertos Fake News, https://www.kaggle.com/c/fake-news/overview

[14] Glazkova, A. A comparison of text preprocessing techniques for hate and offensive speech detection in Twitter. 13, 155 (2023).

[15] F. Wahab, I. Khan, A. Shankar. Fake News Detection Using Machine Learning Techniques. Journal of Computer Networks, Architecture and High Performance Computing. 7. 440-461. (2025).

[16] D. Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and

correlation. *arXiv preprint arXiv:2010.16061 (2020)*.

[17] M. Sokolova, G. Lapalme. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 45 (2009)

[18] Repositorio con la implementación de modelos de detección automática de noticas falsas LSTM y BERT, https://github.com/claupaos/Deteccion-Automatica-de-Noticias-Falsas-

[19] A. Mallik, S. Kumar, Word2Vec and LSTM based deep learning technique for context-free fake news detection. Multimed Tools Appl **,** 83, 919–940 (2024).

[20] P. Dhiman, A. Kaur, Automatic Fake News Identification: A Hybrid CNN-LSTMLogistic Regression Approach, Journal of Information Systems Engineering and Management 10, (2025)